SMC-TR-06-08

AEROSPACE REPORT NO.
TR-2006(1472)-1

Recommended Process for Qualification Testing of Payload Software Prior to Use on Orbit

20 March 2006

Prepared by

S. J. ALVARADO, J. C. CANTRELL and R. L. KLUNGLE
Software Engineering Subdivision
Computers and Software Division

Prepared for

SPACE AND MISSILE SYSTEMS CENTER
AIR FORCE SPACE COMMAND
330 W. Orbital Loop
Los Angeles Air Force Base, CA 90245

Engineering and Technology Group

THIS DOCUMENT CONTAINED BLANK PAGES THAT HAVE BEEN DELETED

THE AEROSPACE CORPORATION
El Segundo, California

20060508023

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

This report was submitted by The Aerospace Corporation, El Segundo, CA 90245-4691, under Contract No. FA8802-04-C-0001 with the Space and Missile Systems Center, 330 W. Orbital Loop, Los Angeles Air Force Base, CA 90245. It was reviewed and approved for The Aerospace Corporation by L. H. Miller, Principal Engineer/Scientist, Space-Based Surveillance Division. Col. Mitchell Heitmann was the project officer for the program.

This report has been reviewed by the Public Affairs Office (PAS) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication. Publication of this report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.

Mitchell L. Heitmann

Col. Mitchell Heitmann
SMC/ISS

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (*DD-MM-YYYY*)	2. REPORT TYPE	3. DATES COVERED *(From - To)*
20-03-2006		

4. TITLE AND SUBTITLE

Recommended Process for Qualification Testing of Payload Software Prior to Use on Orbit

5a. CONTRACT NUMBER
FA8802-04-C-0001

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

S. J. Alvarado, J. C. Cantrell, and R. L. Klungle

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

The Aerospace Corporation
Laboratory Operations
El Segundo, CA 90245-4691

8. PERFORMING ORGANIZATION REPORT NUMBER

TR-2006(1472)-1

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Space and Missile Systems Center
Air Force Space Command
330 W. Orbital Loop
Los Angeles Air Force Base, CA 90245

10. SPONSOR/MONITOR'S ACRONYM(S)
SMC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
SMC-TR-06-08

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report defines the minimum required conditions necessary to qualify payload software prior to launch or use on orbit.

* The recommended approach is to complete full payload system (software, hardware, integration, and external interface) qualification prior to launch.
* If launch will take place before full payload qualification, then
  - Core payload software functionality (i.e., bootstrap startup, spacecraft communications, system upload, hardware/software status reporting, and safing) must be fully qualified before launch.
  - Incremental upload of the software must be planned, architected, designed, and exhaustively tested.
  - Core software must be integrated and tested with all unqualified software using a qualified test environment prior to upload.

In either case, full qualification of both core and basic (i.e., non-core) payload software must be complete before on-orbit use.

15. SUBJECT TERMS

Space systems, software systems, payload software, software testing, qualification testing, on-orbit software

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			John Cantrell
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED		15	19b. TELEPHONE NUMBER *(include area code)* (310)336-2899

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. 239.18

Executive Summary

This report defines the minimum required conditions necessary for qualifying payload software prior to launch or use on orbit.

- The recommended approach is to complete full payload system (software, hardware, integration and external interface) qualification prior to launch.
- If launch will take place before full payload qualification, then
  - Core payload software functionality (i.e., bootstrap startup, spacecraft communications, system upload, hardware/software status reporting, and safing) must be fully qualified before launch.
  - Incremental upload of the software must be planned, architected, designed, and exhaustively tested.
  - Core software must be integrated and tested with all unqualified software using a qualified test environment prior to upload.

In either case, full qualification of both core and basic (i.e., non-core) payload software must be complete before on-orbit use.

Independent verification & validation is a method to reduce risk and increase the reliability of the payload software.

Acknowledgment

The authors would like to thank the following members of the Computer and Software Division for their comments and assistance in preparing this report.

Richard Adams

Douglas Buettner

Lance Diernback

Suellen Eslinger

Leslie Holloway

Larry Jansen

Lee Marvin

Mary A. Rich

Marilee Wheaton

Contents

1. Scope 1
2. Definition of Terms 3
   2.1 Software Qualification Testing 3
   2.2 Payload Software 3
3. Payload Software Functions 5
4. Testing the Payload Software 9
   4.1 Software Qualification Testing Scope 9
   4.2 Testing Environment 10
5. Conclusion 13

References 15

Figure

1 Payload qualification process flow. 7

Table

1. Payload Capabilities 5

1. Scope

This report defines the minimum required conditions necessary for qualifying payload software prior to launch or use on orbit. This minimum set of conditions must be satisfied for Aerospace to recommend launch of or use of payload software on orbit.

2. Definition of Terms

2.1 Software Qualification Testing

Test Requirements For Launch, Upper-Stage, and Space Vehicles[1] states "Software qualification testing verifies that the software meets its specified requirements." Reference 1 goes on to provide several important comments on software qualification testing.

- "Software qualification testing completion criteria shall include the successful execution of software qualification test cases covering, as a minimum, verification of all software requirements under conditions that are as close as possible to those that the software will encounter in the operational environment (e.g., operational flight data constants, operational input and output data rates, target flight hardware configurations);

- verification of all software interface requirements, using the actual interfaces wherever possible or high-fidelity simulation of the interfaces where not possible;

- verification of all software specialty engineering requirements (i.e., supportability, testability, reliability/maintainability/availability, safety, security, as applicable), including in particular verification of software reliability requirements and fault detection, isolation, and recovery requirements;

- stress testing, including worst-case scenario(s); and resource utilization measurement (e.g., CPU, memory, storage, bandwidth)."

The testing and verification above falls into the category of "requirements-based" testing. These tests include full path coverage at the unit level, interface testing in accordance with the applicable Interface Control Documents (ICDs), nominal and off-nominal testing, and stress testing. Full qualification, as used in this report, includes all the above testing and verification efforts performed on the payload hardware and software together with robustness testing,[3] which characterizes how the payload software responds to invalid input parameters.*

2.2 Payload Software

Payload software is a part of flight software, that is, all the software on the satellite while on orbit. Payload software, as the term is used in this report, refers to all software used on orbit by the payload hardware. Such software includes any firmware used to directly control hardware as well as the programs used in any form on any processors included with the payload. Payload software also includes all tables and parameters kept onboard and accessed by the payload software to perform its functions.

* This report does not deal with other methods of software qualification, such as inspection or analysis. While those methods are important, they are not covered in this report.

For the purposes of this report, payload software does not include any functions performed by the spacecraft for its own purposes or on behalf of the payload.

3. Payload Software Functions

Table 1 lists the core and basic functions performed by payload software. Because payload capabilities vary tremendously based on their mission, this set of capabilities must be evaluated for each system to ensure completeness. The terminology used will vary from mission to mission and contractor to contractor, but these capabilities will be present in all payloads. Table 1 does not give a special entry to any software or firmware that *cannot* be uploaded after launch. Such software or firmware must be considered a possible single-point failure, and should be treated accordingly.

The first five capabilities in Table 1 comprise the core set of payload capabilities and are critical for minimum payload operation. If any of these five capabilities fail, the result could range from degraded payload performance to total payload failure. If the five core capabilities have been fully tested and qualified as defined in Subsection 4.1 of this report, it is possible to correct problems in the remaining functions by uploading modified software. To qualify any software, a consistent logical progression of testing, in accordance with good software engineering practice, starting with unit testing to integration testing to system testing, must be completed. These tests must be thorough and include: path and thread testing; off-nominal cases; boundary conditions; stress cases; and robustness testing[3] Robustness testing characterizes how the payload software responds to invalid input parameters.

The last two basic capabilities are important for the payload's mission. However, the core set comprises those functions requiring qualification to allow modification to the core and basic capabilities, whether this modification is due to errors in the software or the desire to have the payload perform functions that were not included in the original design or previous uploads.

The Bootstrap Startup capability is needed to enable ground controllers to restart the processor and bring the payload into a known, safe state. The Bootstrap Startup can be initiated by the spacecraft, the ground controllers, or a cold boot startup initiated by the payload (including the operating system and a restart command on the payload that forces a reboot). It may also perform some initial hardware checks (e.g., memory testing) and reporting of the results to the spacecraft for telemetry downlink. From this known, safe state, ground controllers can direct the next step for the payload. This could be an upload of the operational payload software, a jump to start payload operation if the soft-

Table 1. Payload Capabilities

1	Bootstrap Startup	Core
2	Spacecraft Communications	Core
3	System Upload	Core
4	Hardware/Software Status Reporting	Core
5	Safing the payload	Core
6	Hardware Control	Basic
7	Payload Data Processing	Basic

ware is already onboard, or starting some diagnostic mode if necessary. If the Bootstrap Startup does not operate properly, it will be impossible to put the payload into a known, safe state and may lead to payload mission failure since the ground controllers may be unable to start the operational payload software.

The System Upload capability is necessary to reprogram the payload processor and other reprogrammable devices in the payload, as well as provide updated payload data. This function is essential for remedying onboard software faults or for providing payload software changes required to the existing onboard software over time. This capability decreases the probability that the payload software can place the payload in an unrecoverable state.

The Spacecraft Communications capability includes any communications involving commanding of the payload and receiving the payload response. This report assumes that the spacecraft provides all communications services between the ground and the payload. If the payload cannot communicate with the ground controllers, it will be unable to provide data for downlink to the ground and incapable of receiving ground commands such as uploading the software. If this communications path does not operate correctly, the payload is useless.

The Hardware/Software Status Reporting capability provides health, status, and debug information to the ground. This information is more comprehensive than the information provided during Bootstrap Startup and is intended to provide the ground with sufficient information to troubleshoot problems, allow trending of parameters and the like.

The Safing function ensures that the payload can be brought into a known, safe state that will preserve the payload from the environment in the event of problems. This can be activated either by a ground command or due to a problem in the payload itself. It is included as part of the core set to ensure that the payload can be brought into a known, safe state from any other payload state. Once the payload has entered a Safe Hold state, it is critical that the payload can be commanded to transition out of Safe Hold into a test, operational, or shutdown state.

In summary, these five core capabilities must be fully qualified prior to launch. Without Spacecraft Communications or the Bootstrap Startup, the payload *will* not function, and without System Upload, the payload *may* not function, as it would be impossible to correct any software errors found in the payload software. If the Hardware/Software Status Reporting fails, the ground may not be able to determine whether any of the other functions are operating as intended and may not be able to verify that data coming from the payload is correct. After executing the Bootstrap Startup function, the payload is in a known, safe state. If, after transitioning from this state into a mission state, the payload hardware or software encounters a problem that it is not prepared for, there must be a method to return to a known, safe state. That is the responsibility of the Safing function. If this function fails to operate properly, the payload may be exposed to conditions that degrade its ability to perform the mission. The Safing function can be triggered by payload hardware or software, the spacecraft, or the ground controllers.

While it is not a foregone conclusion that any of the core capabilities will fail without being fully qualified, the risk involved is viewed as unacceptably high. Qualification testing of these core func-

tions ensures that the best effort has been made to eliminate problems that would jeopardize mission success while it is still possible to remedy any such problems. After launch, correcting some errors may prove impossible.

The remaining two basic functions, Hardware Control and Payload Data Processing, comprise the remainder of the payload functions. The Hardware Control capability ensures that the various hardware subsystems of the payload are functioning as required. Payload Data Processing ensures that the data provided by the hardware are processed and prepared for downlink via the satellite bus. As stated previously, if the five core capabilities have been fully qualified, it is possible to correct problems in the remaining functions by uploading modified software. It may also be necessary to modify the software for the core functions. Such modifications to the core set would require a hardware design that allows for the modification of the core software, e.g., storing the core set of functions in field programmable circuitry, and require that the core functions be operating properly, or in some degraded but functional mode that would still permit uploading of software.

As shown in Figure 1, the qualification of flight hardware and simulator software is expected to occur in parallel. The initial software qualification is targeted to achieving full qualification of the core set prior to launch. Once on orbit, only the ***qualified*** payload software can be activated. Operational data will be sent to the ground, generated by the qualified software, for analysis, and, if necessary, improvements made to the software and payload simulator. As new or modified software is introduced, the new system must be fully regression tested prior to upload. This process of software qualification must be iterated until all functionality has been developed and fully qualified before upload.

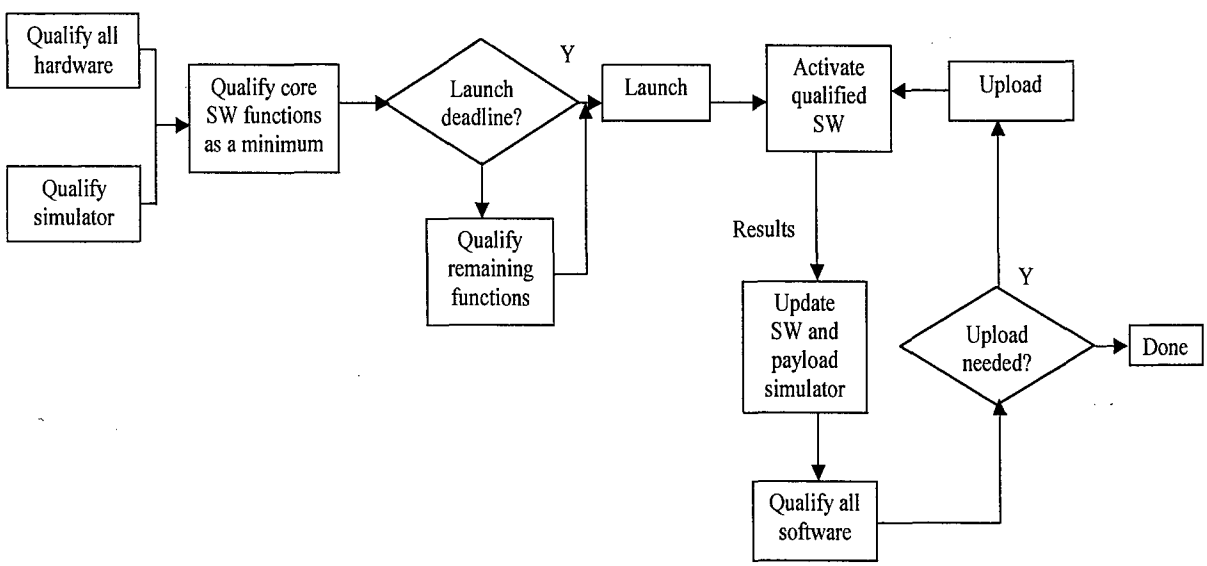

Figure 1 Payload qualification process flow.

4. Testing the Payload Software

4.1 Software Qualification Testing Scope

The following types of testing lead to full qualification of the software:

- Interface (according to the ICDs)
- Nominal (including boundary testing)
- Off-nominal (out of bounds testing)
- Stress
- Robustness
- Full regression (for any modifications)

The various testing types should be applied to the software at the appropriate phase of development. That is, while off-nominal testing would be applied at the unit-testing phase, it is just as suitable at the system level, only with different inputs. Some of these tests may not be possible in certain phases; for example, stress testing does not make much sense at the unit level. Robustness testing is included to cover those testing cases to ensure that *only* qualified software can be activated under any circumstance and that *no* unqualified software can be activated. This verifies that the core set (plus any other qualified software) is adequately decoupled from the remaining, unqualified software.

As new functionality is added to the existing software base, qualification testing of the new functionality must be performed in concert with requalification of the existing functionality. That is, adding or changing software requires not only the full gamut of testing cited above for the new, added software but also that regression testing be performed on both the new or modified software and the existing capabilities to ensure that the new functions do not have a negative effect on the existing, qualified functions. This regression testing must include all the qualification testing previously performed on the core set, and a representative subset of the testing performed on any basic capabilities previously qualified.

Proper documentation of the successful completion of the testing performed during the appropriate phase of software development is an essential element of full qualification testing. Such artifacts demonstrate the development organization's commitment to logical, consistent, suitable software processes; provide the necessary information for maintenance of the software; and increase confidence that the payload software will perform as desired on orbit.

4.2 Testing Environment

Software qualification testing ideally is performed using the actual flight target software and hardware (hardware in the loop) in a configuration as close as possible to the operational configuration. While spacecraft bus, gyro, and other simulators may be used, all capabilities must be tested on real hardware (flight computers in flight configuration), either with the actual flight hardware or engineering flight-like models with well-known and understood deltas to the actual flight software and hardware. These capabilities must also have undergone, and passed, all system-level testing. This system-level testing must include an end-to-end test, including the ground software and hardware to ensure that the payload cannot just communicate with the spacecraft but that this communication extends to the ground as well.

Software qualification testing specifies the use of actual interfaces wherever possible or high-fidelity simulation of the interfaces where not possible. If hardware is not available, simulators must be used. These simulators, together with any hardware in the loop, must be qualified if they are to be used to qualify payload software. The software and hardware qualification testing for any simulator is to ensure that the test system is flight-like with known and well-understood deltas to the actual flight software and hardware.

An engineering model, as defined in this report, comprises as much flight hardware as feasible and is intended to emulate the actual payload and spacecraft interfaces. The engineering model must also include simulators for any subsystems that are not available. A simulator is an implementation of a subsystem that may later be built in hardware or software.

In testing, actual flight hardware is preferable to engineering models, and engineering models are preferable to high-fidelity simulators. While low-fidelity simulators may be used early in the software development and integration phases, such simulators are not considered adequate once the project reaches the software qualification phase. If flight hardware used in ground testing does not provide data adequate for qualification testing of the payload software, either engineering models or high-fidelity simulators must be used.

A testbed is a combination of hardware and software used to simulate the payload and its on-orbit environment. It includes a flight-like payload, the engineering model or a high-fidelity payload simulator, together with the interfaces that the payload will use on orbit. These interfaces may be either hardware or software or a combination of both. It is imperative that the contractor provide a testbed of some type, separate from the testbed used for software development. A good testbed is necessary during the integration and testing phase as it allows testing and software development to proceed in parallel and it permits testing of functionality in ways not convenient on the flight hardware. After launch, the testbed provides a facility for continuing testing, both of newly developed software and for qualification purposes, and anomaly resolution. Before a problem can be solved, it must be diagnosed, and the root cause well understood. The testbed also allows changes to the payload software to be tested thoroughly before being uploaded to the actual payload.

An ideal testbed would contain a flight-worthy payload hardware system and the supporting software. Using a real payload in this role is rarely possible. The use of engineering models is the most likely

second choice. Problems may arise if the engineering model does not use flight-like hardware, for example, a slower processor than the one used on the flight unit. Such a processor will skew performance testing of the payload. The testbed will make use of the various simulations mentioned above. A strong lesson learned is that the overall quality and fidelity of the test environment have a direct impact on software quality, reliability, and risk.

5. Conclusion

This report has defined the minimum required conditions necessary for qualifying payload software prior to launch or use on orbit. The ideal qualification process involves qualifying all payload software before launch following the standard practices described in References 1, 2, and 3. If launch takes place before all payload software is qualified, then the alternate path within the qualification process involves qualifying as much software as possible to include all of the core payload software functionality (i.e., bootstrap startup, spacecraft communications, system upload, hardware/software status reporting, and safing) before launch, activating this core when the payload is on orbit, qualifying non-core payload software on the ground using a qualified payload simulator, and uploading and activating the non-core payload software. Regardless of which path is taken, this qualification process requires that the core and non-core payload software be qualified before being used on orbit. It should also be stated that even fully qualified software could experience anomalies on orbit. The core capabilities must allow for effective troubleshooting from the ground.

All core payload software must be fully qualified before being launched, and all payload software must be qualified before being used on orbit.

References

1. Perl, E., "Test Requirements for Launch, Upper Stage, and Space Vehicles," Aerospace Report No. TR-2004(8583)-1, 31 January 2004.

2. Adams, R. J. et al., "Software Development Standard for Space Systems," Aerospace Report No. TOR-2004(C3909)-3537 Rev A., 11 March 2005.

3. "Software Considerations in Airborne Systems and Equipment Certification," Document No. RTCA/DO-178B, December 1, 1992